



OCI | WE ARE SOFTWARE ENGINEERS.

Database Migration with Grails 3

Grails 101



NIRAV ASSAR – OCI Grails Developer

Introduction

- Nirav Assar
- rails developer since 2009
- DFW area - worked in Financial, retail, Defense, side jobs

Define Database Migration

- Let's Define the term Database Migration
 - what are some interpretations
 - a bit confusing in the broad scope of technology terms
- What are the risks in not having a database migration tool
- Dangers in just using hibernate dbCreate: update

Agenda

- Learn how to use the Grails Database Migration Plugin
 - Live coding examples
 - The plugin is based on Liquibase
 - Discussion of fundamental concepts
- We will achieve the following:
 - Baseline the database for database migration
 - Change a column to nullable
 - Add columns to an existing table
 - Redesign the tables and migrate the existing data -stretch goal

Sample Application and Database

- mysql installation
- Person domain who address information
- Person attributes will evolve
- Simple scaffold UI
- The application is already created, and we will do some dbmigration examples

Configure MySql

- build.gradle - enter runtime dependency
- application.yml

Person Domain Class

- Person: name and age
- PersonController with Scaffold
- insert data in Bootstrap

Install Database Migration Plugin

- `build.gradle` - buildscript dependencies and compile dependencies
- add to `sourceSets` to read changelog

Database Migration in Action

- manages the structure changes made to the db
- automates incremental changes and tracks them

Workflow

- Baseline

1. Define current state of the domain
2. Create the database from the changelog using liquibase
3. Set configuration options in the application to use the database migration plugin

- Develop

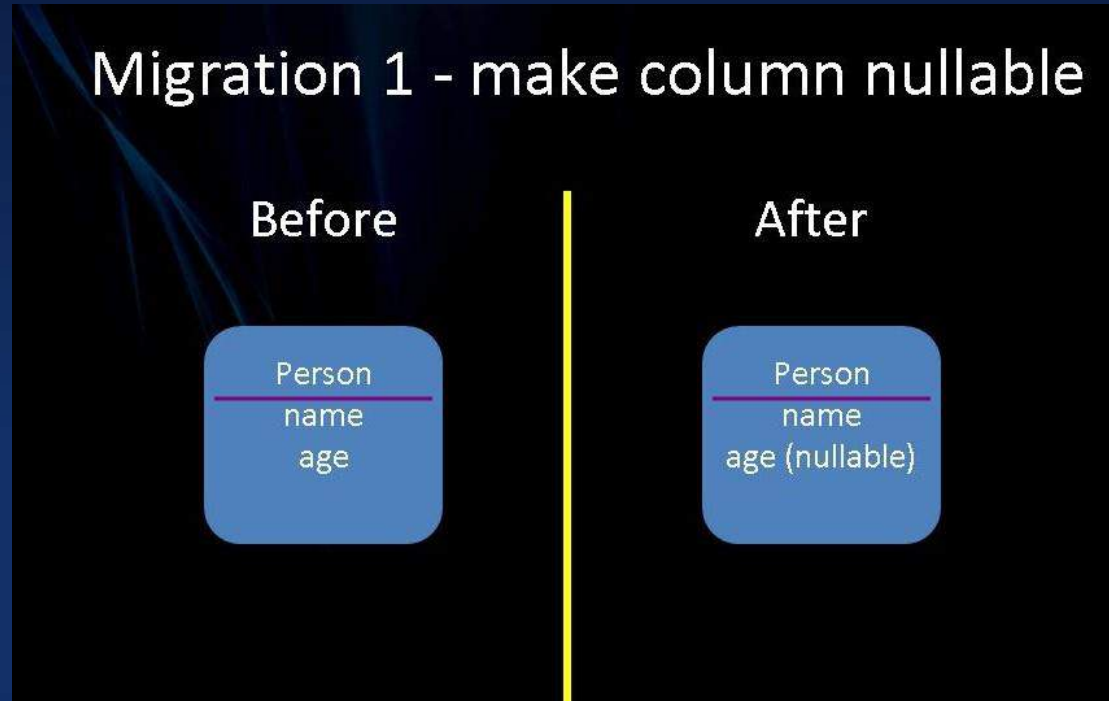
1. Make changes to domain objects
2. Use the plugin to generate changelog additions for the database
3. Update the database using the plugin

Baseline For Database Migration

- application.yml - updateOnStart, updateOnStartFileNames, dbCreate = none
- extract the file to break it into atomic pieces
- generate changelog
- Note DATABASECHANGELOG, DATABASECHANGELOGLOCK tables

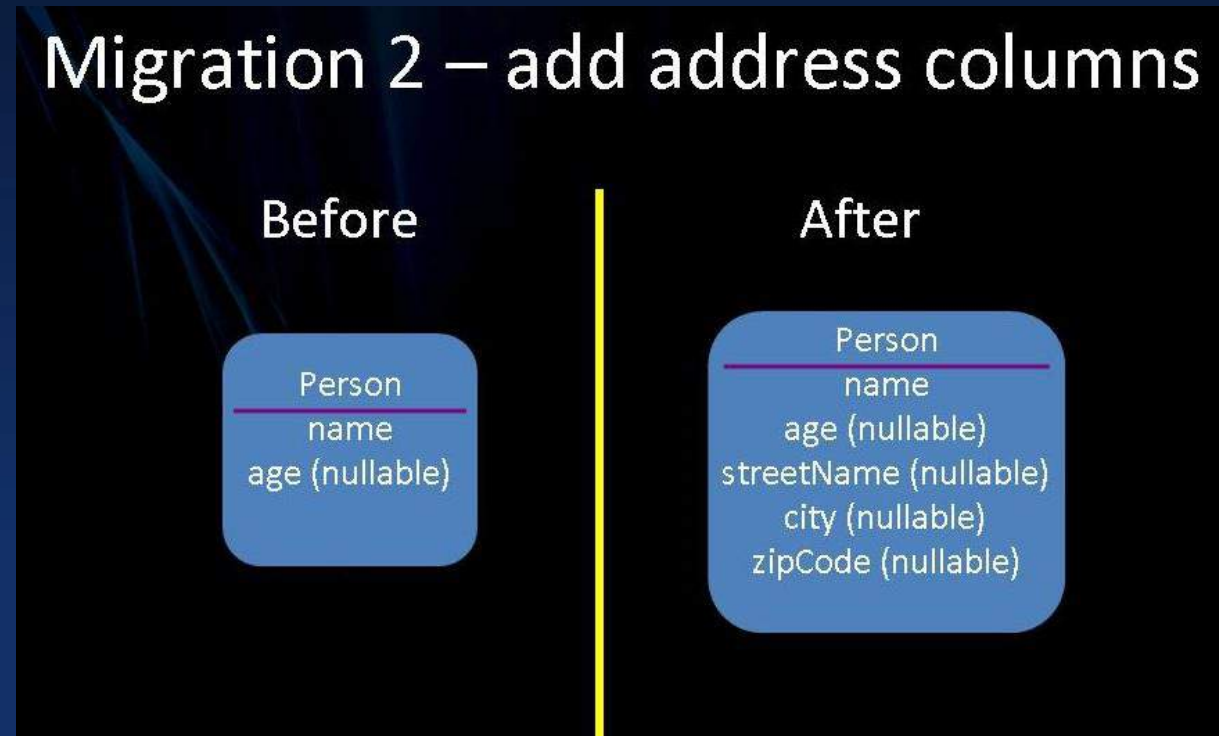
Make Columns Nullable

- Simple change to make age nullable - Person



Add Attributes

- add to Person
- screen shot
- add data



Redesign Tables

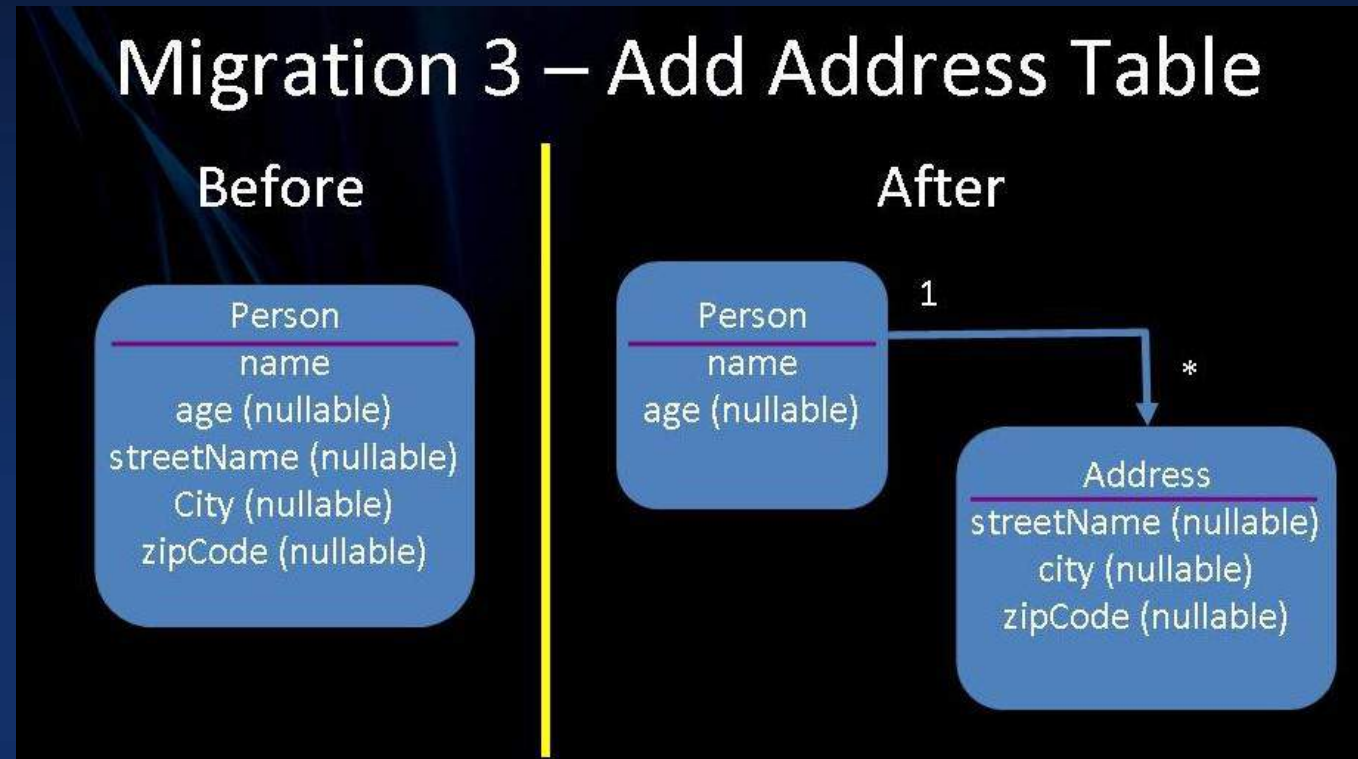
- Person has address information split off into Address domain
- One to Many Relationship arises
- little different then other scenarios
- DON'T forget to add to person table with UI!

Redesign Concerns

- 1. The database table schema definition will change
- 2. Existing data in the table will have to be split between the new database tables created
- 3. We can write custom sql in the changelog files to transfer existing data

Split the Domain

- Person has many Addresses
- create Address UI



Migrate data

- analyze the changeset
- necessary to migrate the data
- only a developer has this knowledge
- liquibase helps you organize
- must occur after creation of Address, but before dropping of Person columns

Review Development Workflow

1. Make changes to Domain objects.
2. Generate the changelog
3. Consider any existing data to migrate.
4. Execute the database migration scripts.

Questions

- Database Migration Guide:
 - <http://guides.grails.org/grails-database-migration/guide/index.html>
- Grails Guides
 - <http://guides.grails.org>
- Nirav Assar
 - assarn@objectcomputing.com

Better APIs with GORM and GRAPHQL

Online Workshop

- April 20th – 9am-12pm
- <https://objectcomputing.com/training/catalog/grails/gorm-graphql-in-depth-workshop>